

**Presentado por:** Manuel Felipe Porras Tascón

**Código:** 201913911

1. Complete la siguiente tabla, con respecto a la creación de threads usando la extensión de la clase Thread y la implementación de la interface Runnable.

Se parecen	Se diferencia
Realizan la misma función de manejar 1 o más threads de forma concurrente.	Se diferencian en la implementación ,ya que cuando se implementa Runnable simplemente se tiene crear la instancia de la clase Thread y dentro de la instancia se tiene que ejecutar el método constructor de la misma clase en la que se está trabajando, mientras que con el extends de Thread simplemente se crea una instancia de la misma clase en la que se está trabajando que extiende de Thread.

2. ¿Al ejecutar el programa, el resultado corresponde al valor esperado?

Si corresponde al valor esperado de 10.000.000.

3. ¿Al ejecutar el programa, el resultado corresponde al valor esperado?

No corresponde al valor esperado, porque los threads se superponen entre si y el contador no cuenta correctamente los 10000 números que debería contar por thread. Por otro lado, si por lo menos se incurre en utilizar el método sleep adentro del mismo for donde se crean e inicia cada thread con por lo menos 5 milisegundos si funcionaria y contaría correctamente.

- 4.

Ejecución	Valor Obtenido
1	9970000
2	6500000
3	4730000
4	9810000
5	3760000

5. ¿Hay acceso recurrente en alguna variable compartida?

Si, en la variable contador, ya que esta variable es accedida por todos los Threads y es modificada 10000 por cada Thread. Específicamente en el método run donde cada thread modifica 10000 la variable contador.

- 6.

Ejecución	Valor Esperado	Valor Obtenido
1	94128	94128
2	98992	98417
3	95693	80687
4	96762	78714
5	88108	77429

**Presentado por:** Manuel Felipe Porras Tascón

**Código:** 201913911

7. ¿Hay acceso concurrente a alguna variable compartida? Si es así, diga en dónde

Hay un acceso concurrente en el método run, específicamente en la variable mayor cuando se realiza la comparación con el mayor Fila, ya que los threads traen el mismo dato de la variable mayor de forma concurrente.

8. ¿Puede obtener alguna conclusión?

Se puede concluir en este laboratorio, que cuando se trabajan con varios threads de forma concurrente se tiene que tener en cuenta que tienen que existir un orden de ejecución para que no haya una superposición de uso de datos durante las ejecuciones del programa, más que todo cuando los threads comparten la misma fuente de datos.